Techset Composition Ltd, Salisbury
Doc: {IEE}IPR/Articles/Pagination/IPR20140421.3d

IET Image Processing

Review Article

IET Journals
The Institution of Engineering and Technology

Logarithmical hopping encoding: a low computational complexity algorithm for image compression

ISSN 1751-9659
Received on 29th January 2014
Accepted on 10th December 2014
doi: 10.1049/iet-ipr.2014.0421
www.ietdl.org

Jose Javier García Aranda[1] ✉, *Marina González Casquete*[1], *Mario Cao Cueto*[2], *Joaquín Navarro Salmerón*[2], *Francisco González Vidal*[2]

[1]*Video Architecture Department, Alcatel-Lucent, Madrid, Spain*
[2]*Departamento de Ingeniería de Sistemas Telemáticos (DIT), Universidad Politécnica de Madrid (UPM), Madrid, Spain*
✉ *E-mail: jose_javier.garcia_aranda@alcatel-lucent.com*

Abstract: LHE (logarithmical hopping encoding) is a computationally efficient image compression algorithm that exploits the Weber–Fechner law to encode the error between colour component predictions and the actual value of such components. More concretely, for each pixel, luminance and chrominance predictions are calculated as a function of the surrounding pixels and then the error between the predictions and the actual values are logarithmically quantised. The main advantage of LHE is that although it is capable of achieving a low-bit rate encoding with high quality results in terms of peak signal-to-noise ratio (PSNR) and image quality metrics with full-reference (FSIM) and non-reference (blind/referenceless image spatial quality evaluator), its time complexity is $O(n)$ and its memory complexity is $O(1)$. Furthermore, an enhanced version of the algorithm is proposed, where the output codes provided by the logarithmical quantiser are used in a pre-processing stage to estimate the perceptual relevance of the image blocks. This allows the algorithm to downsample the blocks with low perceptual relevance, thus improving the compression rate. The performance of LHE is especially remarkable when the bit per pixel rate is low, showing much better quality, in terms of PSNR and FSIM, than JPEG and slightly lower quality than JPEG-2000 but being more computationally efficient.

1 Introduction

There are three main concepts that set the limits for image compression techniques: image complexity [1], desired quality and computational cost. This paper presents logarithmical hopping encoding (LHE) algorithm, a computationally efficient algorithm for image compression.

The proposed algorithm relies on Weber–Fechner law, which states that subjective sensation is proportional to the logarithm of the stimulus intensity [2]. LHE applies this law to prediction errors instead of the stimulus itself (in this case the original luminance and chrominance signals). More concretely, LHE estimates a luminance and chrominance prediction for each pixel (using the surrounding pixels) and then encodes the prediction error using a set of logarithmically distributed and dynamically adjusted values. This procedure is performed in the space domain, avoiding the need of any costly transformation to the frequency domain, and therefore reducing the computational complexity.

This approach can be enhanced by including a pre-processing stage to estimate the perceptual relevance of the image blocks. As will be explained later, the perceptual relevance estimation can be obtained, in an efficient manner, using the output codes of the logarithmical quantiser. This pre-processing stage allows the algorithm to perform region of interest (ROI) coding [3].

The remaining of this paper is organised as follows. Section 2 surveys the most relevant work related to the proposed algorithm. Section 3 describes the structure and the workflow of LHE. In Section 4, an enhanced version of the algorithm is presented. Section 5 shows the main results that have been obtained in the evaluation stage of the algorithm. Finally, Section 6 summarises the main contributions of this paper and outlines the future lines of work.

2 Related work

LHE can be defined as a spatial domain image compression algorithm. In the literature of image compression, spatial domain-based algorithms have been extensively studied.

In [4], a spatial domain image compression algorithm is proposed. This algorithm encodes the difference between the minimum pixel value of an $m \times n$ pixel block and the current pixel. For each block, an 11 bits header is included in order to represent the minimum value of the block (8 bits) and the number of pixels required to encode the pixel difference with respect to the minimum (3 bits).

The authors of [5] present a modified approach to the previous algorithm where the final number of bits is reduced significantly by reducing the overhead bits. In [6], a variation of the previous algorithm that encodes the difference between adjacent pixels is proposed.

In [7], a logarithmic function is used as a pre-processing stage for an image compression algorithm. This algorithm comprises four stages: logarithmic transform, neighbouring difference, repeat reduction and Huffman encoding. In this paper, the logarithmic function is used to reduce the range of the difference between neighbouring pixels.

LHE also has similarities to ADPCM (adaptive differential pulse-code modulation) [8]. ADPCM uses an adaptive predictor and an adaptive quantiser. The quantiser levels for a given pixel are generated by scaling the levels used for the previous pixel by a factor that depends on the reconstruction level used for the previous pixel. ADPCM dynamically adapts the quantiser step size to the input signal, and the set of possible unary codes is linearly distributed for each sample. However, LHE unary codes are logarithmically distributed for each sample. This change in step distribution provides better results than ADPCM. For example, according to [8], Lena image at 1.2 bpp encoded with ADPCM provides a peak signal-to-noise ratio (PSNR) value of 30.24 dB whereas LHE provides 39.1 dB.

LOCO-I [9] is the algorithm at the core of the ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS.LOCO-I uses the prediction of samples based on a finite subset of available past data and the context modelling of the prediction error. The purpose of this context modelling is to exploit high order structures, for example, texture patterns, by

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

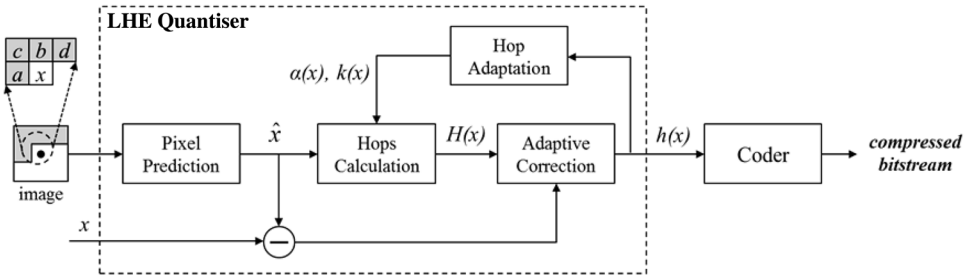

Fig. 1 *LHE basic: block diagram*

analysing the level of activities, such as smoothness and edginess of the neighbouring samples. This context modelling provides a probabilistic model for the prediction residual (or error signal), which can be efficiently used in combination with Golomb-Rice codes. LHE uses a similar approach, but focused on lossy image compression and taking into account the logarithmical nature of human perception.

3 LHE: basic algorithm

The basic algorithm of LHE is based on the prediction of colour space values (e.g. YUV) of each pixel from the previous ones. The errors of the predicted values are encoded using a set of logarithmically distributed possible values of luminance and chrominance, which are called hops. The main blocks of the basic algorithm of LHE, grouped as LHE quantiser, are depicted in Fig. 1. Detailed information about these blocks is provided in the following subsections.

3.1 Pixel prediction

LHE uses the YUV colour space to represent the pixel information. YUV is defined in terms of one luminance value (Y) and two chrominance components (UV). The human eye has fairly little spatial sensitivity to colour, thus luminance component has far more impact on the image detail than the chrominance. For a given pixel, LHE predicts each colour component (Y, U or V) as the average of the colour component of the top (b) and left (a) pixels, as in the following equation

$$\hat{x} = \frac{a+b}{2} \tag{1}$$

The prediction of each colour component should be computed individually. Therefore, for a given pixel x, $\hat{x}$ represents the predicted value of the luminance (Y) or chrominance (UV). In the remaining of this section, luminance will be used as an example of the three colour components.

As the predictions of the pixels depend on the previous ones, the first pixel of the image is not processed by the LHE quantiser. Thus, its colour components are included uncompressed in order to allow the decoder to process subsequent pixels (see Section 3.5 'LHE decoding').

3.2 Logarithmical hops

As aforementioned, LHE encodes the errors of the predicted colour components from a set of possible logarithmically distributed values (called hops) for each pixel, $H(x) = \{h_{-N}, h_{-(N+1)}, ..., h_{-1}, h_0, h_1, ..., h_{N-1}, h_N\}$. The null hop h_0 means that the error associated with the predicted colour component is lower than the one achieved by a different hop value. The smallest positive and negative hops, h_1 and h_{-1}, are not logarithmically assigned. LHE algorithm adjusts automatically, within a certain range, the value of the hops h_1 and h_{-1} for each pixel depending on the previously encoded pixel through the parameter $\alpha(x)$, which is described in Section 3.4 'hop adaptation'. The first time LHE is executed, an initial fixed value for the parameter α is used, for example, $\alpha = 8$.

The following equation details the different values of the set of hops $H(x)$ for a given pixel

$$h_i = \begin{cases} 0, & \text{if } i = 0 \\ \alpha(x), & \text{if } i = 1 \\ -\alpha(x), & \text{if } i = -1 \\ h_{i-1} \cdot (255 - \hat{x}/k(x))^{1/k(x)}, & \text{if } i > 1 \\ h_{i+1} \cdot (\hat{x}/k(x))^{1/k(x)}, & \text{if } i < 1 \end{cases} \tag{2}$$

The image compression rate of LHE depends on the number of hops is considered $(2N+1)$, the smallest non-null hops (h_1 and h_{-1}, defined by the parameter $\alpha(x)$) and the parameter $k(x)$. The higher the cardinality of H, the lower the compression rate of LHE. The parameters $\alpha(x)$ and $k(x)$ are responsible for the compactness of the set of hops $H(x)$ for a given pixel.

Different values of $k(x)$ allow expanding and shrinking the range covered by the set of logarithmical hops. In image areas where there are high component fluctuations, a low value of $k(x)$ covers the maximum range and provides better results. On the other hand, in soft detailed areas, a high value of $k(x)$ shrinks the set of hops, gaining more accuracy for small changes on colour component. The value of $k(x)$ is determined locally, at each pixel, taking into account the set of surrounding hops

$$k(x) = f(h(a), h(b), h(c), h(d)) \tag{3}$$

For each combination of hops corresponding to the pixels a, b, c and d (pixel positions are shown in Fig. 1), there is an optimal value for $k(x)$, which minimises the error when a new hop for x is chosen. Although a formula could be defined, a pre-calculated table of optimal $k(x)$ values can be generated testing over all pixels from all images from an image database, and therefore setting the best values for any type of image. This strategy avoids deducing the 'best logic' for the formula and therefore simplifies the problem.

3.3 Adaptive correction

The adaptive correction module takes into account two parameters: the set of possible hops $H(x)$, computed in the previous module, and the error associated to the predicted colour component, e.

$$e = x - \hat{x} \tag{4}$$

The output of this module is the hop $h(x)$ from the set $H(x)$, that is, $h(x) \in H(x)$, that is closer to the above described error. In other words, the hop $h(x)$ is the quantised error made by the LHE algorithm in the colour component prediction $\hat{x}$. This hop $h(x)$ is the output of the LHE quantiser

$$h(x) = \arg_{h_i} \min(|h_i - e|), \quad h_i \in H(x) \tag{5}$$

In the particular case that there are two different hops with the same

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

Table 1 Statistical compression for five hops

Quantised error (Hop)	Code, bits
h_0	1
h_1	01
h_{-1}	001
h_2	0001
h_{-2}	00001

distance to the error e, the hop with the smaller value is chosen. The reason behind this approach is that in statistical compression of images, smaller codes are assigned to small hops (see Section 3.5 'coder')

$$\text{If } \exists(h_j, h_k) \in H(x) \Big| \big|h_j - e\big| = \big|h_k - e\big| \Rightarrow h(x) = h_i \Big| i = \min(|j|, |k|) \tag{6}$$

3.4 Hop adaptation

Once the quantised error of the actual pixel is assigned, the hop adaptation module updates the parameter $\alpha(x)$, which has the same absolute colour component value as the smallest non-null hops h_1 and h_{-1}, for the next pixel. The parameter $\alpha(x)$ varies within a certain fixed range $[\alpha_{\min}, \alpha_{\max}]$, for example, [4, 8]. According to (2) the parameter $\alpha(x)$ is used for computing the set of possible hops H of the next pixel. As aforementioned, an initial start value for the parameter α is fixed for the first pixel encoded by LHE, for example, $\alpha = 8$.

The adjustment of the value α is based on the following rules:

- If the assigned hops of two consecutive pixels $\{h(x–1), h(x)\}$ are small, that is, they are either null hops h_0 or the smallest non-null hops $\{h_{-1}, h_1\}$, then the updated value $\alpha(x)$ becomes one unit smaller than the smallest positive non-null hop h_1, up to a certain minimum given by $\alpha_{\min}$

$$\text{If } \{h(x-1), h(x)\} \in \{h_{-1}, h_0, h_1\} \Rightarrow \alpha(x) = \max(h_1 - 1, \alpha_{\min}) \tag{7}$$

- If the quantised error $h(x)$, assigned in the previous module, is different to the null hop or the smallest non-null hops $\{h_{-1}, h_1\}$, then the updated value α is set to its maximum $\alpha_{\max}$

$$\text{If } h(x) \notin \{h_{-1}, h_0, h_1\} \Rightarrow \alpha = \alpha_{\max} \tag{8}$$

3.5 Coder

The coder module translates the quantised hops of all pixels into a compressed stream of bits. One possible approach for the compression technique can be based on the existing redundancy of images across its axes, that is, any pixel is generally similar to the previous one, and thus small hops are more frequently assigned. Although different compression techniques can be applied, this paper recommends the use of Huffman coding algorithm. It has variable-length codes for defining the quantised errors (called hops) based on its frequency of appearance.

However, analysis over two image databases [10, 11] reveals that the smallest hops are assigned in more than 90% of pixels. Therefore, in order to avoid the frequency analysis and enable real-time encoding, an effective statistical compression of hops can be achieved by assigning the smaller codes to the smaller hops. Table 1 shows an example of a LHE statistical coder with five hops codes.

3.6 LHE decoding

The LHE decoder performs similar operations as in the LHE quantiser but in the reverse order. The following lines described the phases of the LHE decoding process for a given pixel x. It should be noted that previous pixels to x has been already decoded and therefore the value of the parameter $\alpha(x)$ for this pixel has been already computed.

1. The binary stream is translated into symbols, in this case, into a certain hop value $h(x)$.
2. The current predicted pixel $\hat{x}$ is computed as the average of the colour components of the top and left pixels, as in (1).
3. Given the value of $\hat{x}$ and α, the set of hops $H(x)$ for this pixel are computed by following (2). At this moment, the colour component value of $h(x)$ is known.
4. The decoded colour component x' is computed as follows

$$x' = \hat{x} + h(x) \tag{9}$$

5. Finally, the new value for the parameter α is computed, following the rules described in Section 3.4 hop adaptation.

As aforementioned, the first pixel colour components are included in raw-format in the binary stream, in order to enable the decoding process of the subsequent pixels.

Fig. 2 shows the process of encoding and decoding a figure with the basic LHE algorithm. The LHE quantiser assigns a symbol to each pixel (in this case, nine hops are considered). Following the above described steps, the hop symbols are decoded as pixel colour components.

4 LHE: enhanced algorithm

4.1 Motivation

In [3, 12], a method known as a region of interest coding is introduced. The main idea behind ROI coding is to segment an

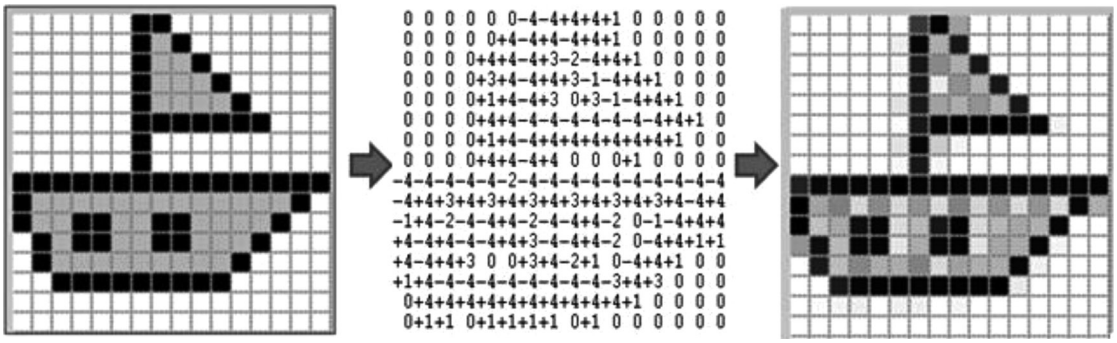

Fig. 2 *LHE encoding and decoding examples*

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

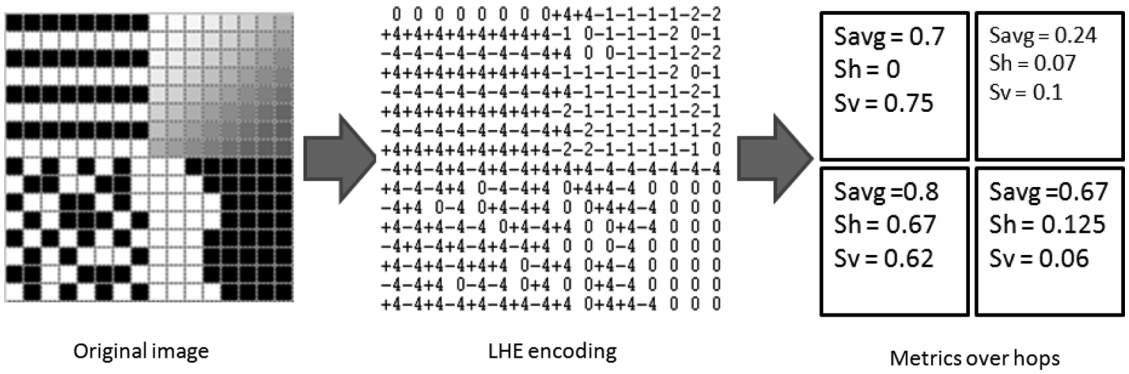

Fig. 3 *Perceptual relevance metrics*

image into an ROI and the background. If the ROI is coded with higher fidelity than the background, a high compression ratio with good subjective quality can be achieved. ROI coding relies in the fact that the background is less perceptually relevant than the ROI, so the coding errors made in the background are more likely to remain unnoticed than if those errors are made in the ROI.

Analogously, LHE adopts the idea of the perceptual relevance of the different regions of an image, but instead of trying to distinguish between an ROI and the background, LHE generalises the ROI concept, by evaluating the perceptual relevance of each block (8 × 8 pixels) of the image and encoding each of these blocks accordingly.

4.2 Perceptual relevance evaluation

Intuitively, the perceptual relevance of an image block can be defined as a measure of the importance of the block regarding to the complete image, as perceived by a human viewer [13]. This subsection explains how the perceptual relevance is estimated in the LHE algorithm.

Throughout the development of the LHE algorithm, we realised that LHE-quantised luminance, that is, the output of the LHE quantiser when using luminance as input, can be used as an estimator of the perceptual relevance of an image block. Three metrics have been defined to estimate the perceptual relevance of each block:

- S_{avg}: absolute hop index average (normalised to 0–1). This metric gives a measurement of the size of the hops. A value close to 1 indicates high luminance/chrominance fluctuations, and therefore, it suggests a complex region such as fur or sea foam and so on. It is equivalent to the entropy measurement.
- S_h: number of changes of hop index's sign when scanning the symbols horizontally (normalised to 0–1). A value close to 0 indicates that the block has low information in the horizontal direction.
- S_v: number of changes of hop index's sign when scanning the symbols vertically (normalised to 0–1). A value close to 0 indicates that the block has low information in the vertical direction.

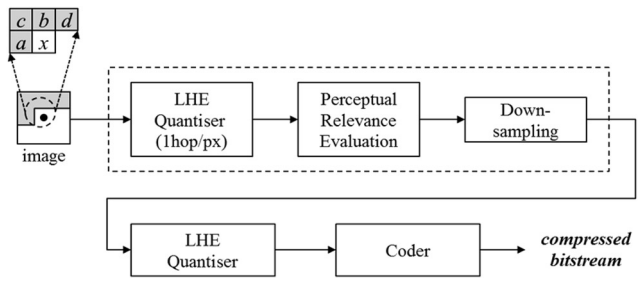

Fig. 4 *Enhanced LHE algorithm*

In Fig. 3, these metrics are calculated over an example image. As can be seen in the example, blocks containing low information at the horizontal direction, have a low S_h value. This allows detecting when a block can be down sampled horizontally. These metrics also make possible the distinction of complex regions (which have high S_{avg}, S_h and S_v values) from soft regions (with low S_{avg}, S_h and S_v values) and edges (high S_{avg} value but low S_h or S_v depending on the edge direction).

The following subsections explain two methods (based on the perceptual relevance metrics) aimed at improving the performance of the basic LHE algorithm. The architecture of the enhanced algorithm is depicted in the following image (Fig. 4). Q2

The enhanced version of the LHE algorithm uses the aforementioned perceptual relevance metrics as input for a new module: downsampling. This module is aimed to reduce the amount of information to be coded, improving the compression rate while maintaining the subjective quality. This module is described in more detail in the following subsection.

4.3 Downsampling

The downsampling module uses the perceptual relevance evaluation to reduce the information to be encoded. For each block, it evaluates the perceptual relevance metrics defined in the previous subsection and it decides if the block can be resized, thus reducing the number of pixels that have to be processed.

The decision to resize an image block depends on a set of thresholds that are applied to S_{avg}, S_h and S_v. More concretely, six thresholds are defined, a maximum and a minimum threshold for each perceptual relevance metric. Depending of the actual value of the perceptual relevance metrics with respect to the thresholds, the downsampling module can estimate the type of content of the image block, and its suitability to be resized.

Table 2 summarises the detection rules that are applied to identify Q3 the type of content of an 8 × 8 block and the resizing strategy that is

Table 2 Downsampling strategies

Type of content	Detection rule[a]	Resizing strategy	Binary code
plain luminance	Savg↓ and Sh↓ and Sv↓	4 × 4 pixels (vertical and horizontal)	11
gradated or soft details	Savg↓ and Sh↓	8 × 4 pixels (horizontal)	01
	Savg↓ and Sv↓	4 × 8 pixels (vertical)	10
strong and fuzzy details, for example, hair	Savg↑ and Sh↑	8 × 4 pixels (horizontal)	01
	Savg↑ and Sv↑	4 × 8 pixels (vertical)	10
other	none threshold is exceeded	no resizing	00

[a]↓ = minimum threshold exceeded and ↑ = maximum threshold exceeded

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

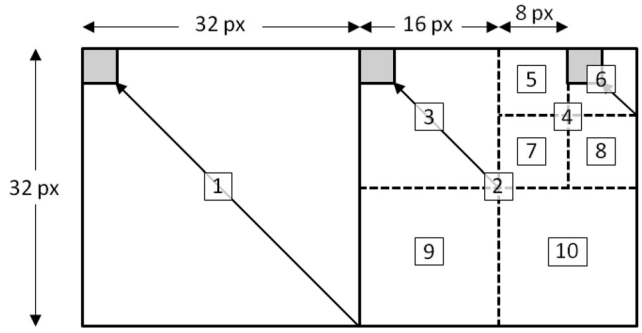

Fig. 5 *Recursive downsampling procedure*

applied in each case. It also shows the binary code that is assigned to each block in order to identify how the block has been resized (vertically, horizontally or both).

Once the image blocks have been analysed (and resized) by the downsampling module, they will be used as input for the LHE quantiser. As can be seen, the main advantage of the downsampling process is that a certain amount of the 64 pixels blocks will be replaced by resized versions of 16 or 32 pixels.

The threshold selection establishes a trade-off between image quality and compression rate. If the thresholds are very restrictive (maximum and minimum thresholds close to 1 and 0, respectively), more quality (and less compression rate) will be achieved. More details about the threshold effect will be given in Section 5.

In order to take the maximum advantage of the downsampling module, a recursive procedure, using different block sizes, can be used. Let 'n' be number of iterations of this recursive procedure. First, the image is divided into macroblocks of $2^{2+n} \times 2^{2+n}$ pixels. For each of these macroblocks, the perceptual relevance metrics are computed, and the resizing rules are checked. If the macroblock can be resized (according to the aforementioned rules) it will be resized and the next macroblock will be processed. If the macroblock cannot be resized, then it is divided into four blocks and the previously described downsampling technique is applied recursively to each of these inner blocks. This recursion is applied while the block size is equal or bigger than 8×8 pixels. The calculation of the perceptual relevance metrics for a macroblock does not require additional processing, because the additive nature

Fig. 6 *JPEG, LHE and JPEG2000 comparison*

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

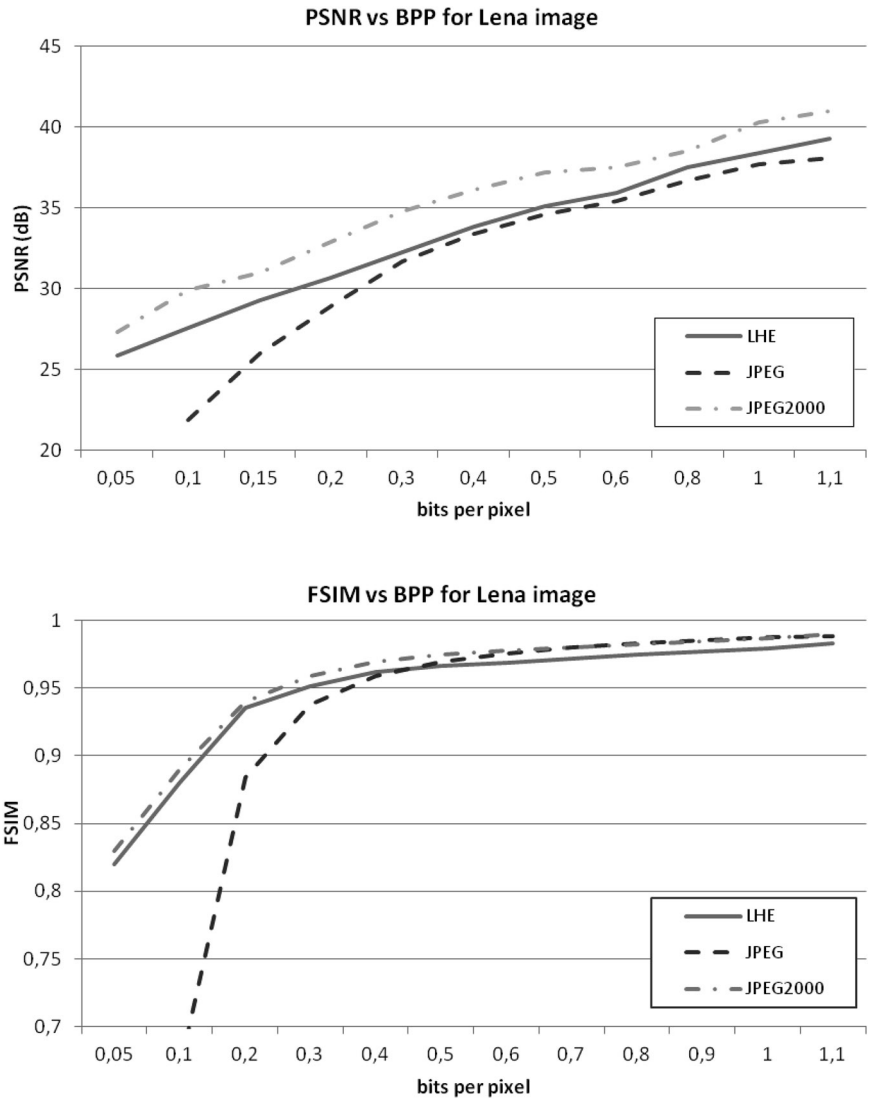

Fig. 7 *PSNR and FSIM R-D diagrams for 'Lena' image*

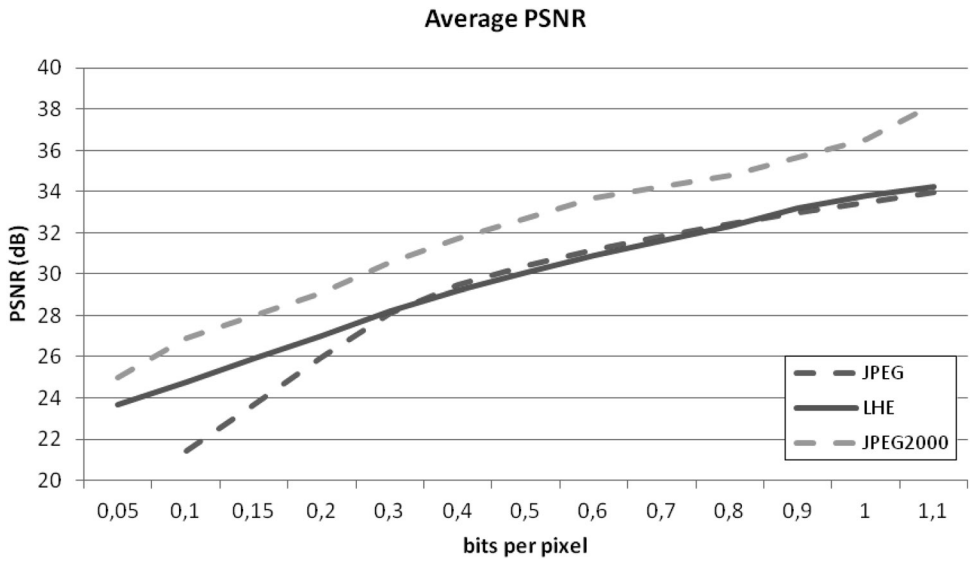

Fig. 8 *Average PSNR diagram*

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

```
//Pixel Prediction
x̂ = (a + b)/2

//Best Hop Selection
error_min = |x − x̂|
hop_selected = 0

//Positive Hops
if (x > x̂) then
    foreach (h_i in h_1 ... h_4) do  //hops can be precomputed
        error = |x − h_i |
        if (error < error_min) then
            hop_selected = i
            error_min = error
        else
            break
        end
    end

//Negative Hops (same loop for h_−1 ... h_−4)
else
    ...
end
```

Fig. 9 *LHE algorithm*

of these metrics allows them to be computed by adding the corresponding metrics of the macroblock inner 8×8 blocks. The perceptual relevance metrics for each 8×8 block are given by the perceptual relevance evaluation module, so the computational overhead in the downsampling module is low.

The following figure shows an example where the recursive downsampling procedure is applied to a 32×64 pixels image, using $n = 3$ levels of iterations (Fig. 5).

The numbers enclosed in boxes represent the processing order for each block and macroblock. As can be seen, in the first place, the 32×32 macroblock on the left is processed. As it can be resized, no further processing is required. Next, the 32×32 macroblock on the right is processed. This macroblock cannot be resized so its 16×16 pixels inner macroblocks will be processed. The first 16×16 macroblock (on the top and the left) can be resized, so the algorithm continues with the next one. The second 16×16 macroblock cannot be resized, so it is required to process its 8×8 inner blocks. This process continues until the complete image has been analysed.

In the decoder side, the downsampling procedure can be easily reverted by applying an interpolation algorithm over the downsampled version of the decoded blocks in order to restore their original size.

LHE does not specify which downsampling or interpolation technique should be used. However, the selected technique will have an effect over the performance and the quality achieved by LHE.

5 Experimental results

The following experimental results have been obtained by applying LHE algorithm with nine hops and by using pixel averaging downsampling and bilinear interpolation.

5.1 Image quality

LHE has been tested using two image databases: Kodak lossless true colour image suite [10] and USC-SIPI image database (miscellaneous volume) [11]. LHE provides, in most cases, a good objective quality (PSNR) but the even better subjective quality, because bigger errors are located at pixels with strong contrast with surrounding ones, where subjective quality impact is minimised.

Edge information is vitally important in the perception of images [14]. However, this information is usually distorted when the image is encoded using DCT (discrete cosine transform) or other frequency-based technique [15, 16]. Fig. 6 shows a subjective quality comparison between JPEG, LHE and JPEG2000. It can be seen that the LHE edges are cleaner than JPEG edges. Furthermore, LHE noise has less impact on the subjective quality compared to typical DCT-based algorithms noise because of the lack of visible artefacts at block boundaries. JPEG2000 obtains higher PSNR values and slightly better subjective quality than LHE. At low bitrates the most important LHE degradation is because of the strong downsampling performed by the encoder in certain images areas, which may lead to blurriness when they are interpolated by the decoder. Despite the aforementioned effect, in terms of subjective quality, LHE is closer to JPEG2000 than JPEG.

Fig. 7 curves are the PSNR and FSIM rate–distortion (*R–D*) diagrams for Lena image. LHE performs better than JPEG at low bitrates and provides quite similar quality at high bitrates. Regarding FSIM, LHE follows JPEG2000 trend.

Fig. 8 shows the average PSNR using all the images contained in the Kodak lossless true colour image suite and the USC-SIPI image database. Analogously to the case of Lena image, for all the images of the databases, LHE outperforms JPEG at low bit rates.

To provide more evidences of the LHE quality performance the following figure shows a comparison between LHE, JPEG and JPEG2000 using blind/referenceless image spatial quality evaluator (BRISQUE) [17], a metric that evaluates the loss of 'naturalness'

Table 3 LHE complexity (linear) without parallelisation

Image resolution, px	184*k* (429 × 429)	414*k* (720 × 576)	921*k* (1280 × 720)	2073*k* (1920 × 1080)
quantisation time, ms	2.8	6.7	15	29.49

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

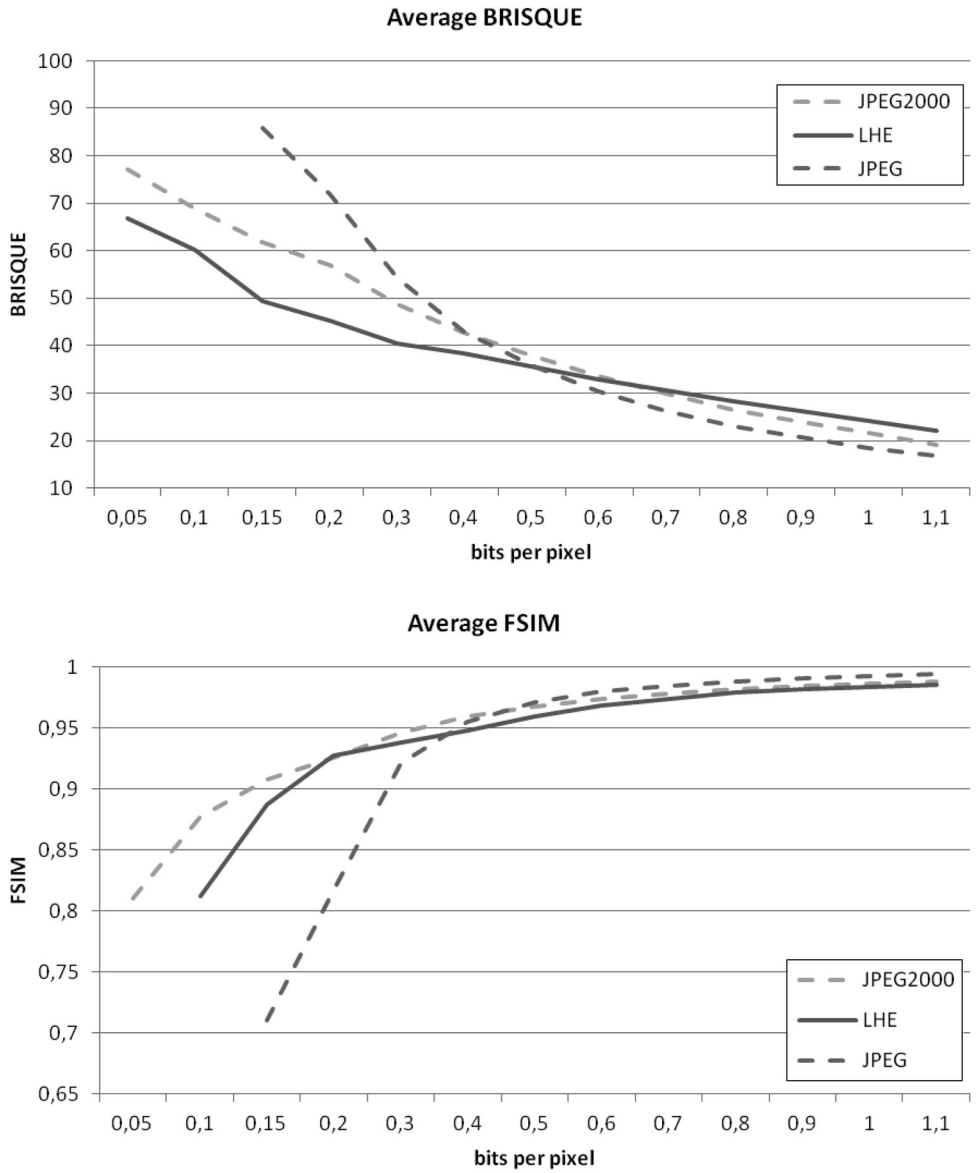

Fig. 10 *LHE parallel processing of N × N pixels in 2N–1 steps*

in the image as a consequence of the compression process; and FSIM [18], a full reference image quality metric consistent with subjective evaluations. It should be noted that quality and BRISQUE are inversely proportional while quality and FSIM are directly proportional.

5.2 Algorithm performance

One on the main advantages of LHE is its simplicity. LHE time complexity is $O(n)$ and its memory complexity is $O(1)$. A simplified implementation of the LHE algorithm for nine hops is described in Fig. 9.

It can be observed that LHE only requires a constant number of basic operations per pixel (assignments and comparisons), that is, its complexity is $O(n)$. In contrast, DCT and DWT (discrete wavelet transform) have $O(n \cdot \log n)$ or higher computational complexity [19]. This advantage makes LHE an extremely fast procedure. Our Java prototype (without parallelisation) tested on Intel i5-3320@2.6Ghz achieves the quantisation times that are shown in Table 3, confirming the linear complexity of LHE.

Using parallelisation, the encoding time can be significantly reduced. LHE allows the parallel processing of those pixels which have the top and left pixels processed (needed for colour components prediction). Given an image of $N \times N$ pixels, the complete parallel process will take $2N$–1 steps. In Fig. 10 pixels are labelled with a number. Pixels labelled with the same number can be encoded in parallel (in the same step).

This parallelisation strategy can be applied to blocks instead of pixels. Thus, the enhanced LHE algorithm can take benefit from parallelisation, where every block is downsampled at different ratio depending on its perceptual relevance metrics. In this case, the encoding of the blocks is made in the same order as depicted in Fig. 11. This figure refers to pixels but the same parallelisation strategy can be applied to blocks composed of $M \times M$ pixels.

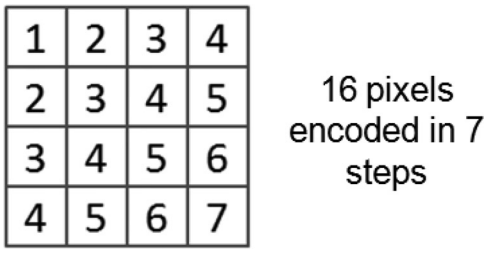

Fig. 11 *Image quality metrics: non-reference (BRISQUE) and full-reference (FSIM)*

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

6 Conclusions

LHE is a lossy compression algorithm suitable for static images based on adaptive logarithmical quantisation. The main advantages of the algorithm are its low computational complexity $O(n)$ and its performance in terms of image quality, specially at low bitrates. Two main contributions make possible this performance. In the first place, LHE proposes a logarithmical quantisation of the error between pixel colour component predictions and the actual value of such components. This quantisation is based on Weber–Fechner law and it has been proven as a linear and quality effective compression procedure. In the second place, a downsampling strategy, based on perceptual relevance metrics (calculated using LHE-quantised luminance), provides a fast procedure to protect the image ROIs, optimising the overall image quality. Furthermore, the algorithm design supports the parallel processing of different image blocks, thus reducing the encoding time.

The results of this paper point to several interesting directions for future work:

- Image quality improvement based on more complex pixel prediction, downsampling strategies and interpolation techniques.
- Modification of LHE algorithm for lossless image compression.
- Application of LHE approach in the time-domain for video and audio compression.

7 Acknowledgment

The authors wish to thank the Spanish Science & Tech Ministry which funds this research through 'INNPACTO' innovation program IPT-2011-1683-430000.

8 References

1 Lloyd, S.: 'Measures of complexity: a nonexhaustive list', *IEEE Control Syst. Mag.*, 2001, **21**, (4), pp. 7–8
2 Jayant, N., Johnston, J., Safranek, R.: 'Signal compression based on models of human perception', *Proc. IEEE*, 1993, **81**, (10), pp. 1385–1422
3 Cruz, D.S., Ebrahimi, T., Larsson, M., Askelof, J., Cristopoulos, C.: 'Region of Interest coding in JPEG2000 for interactive client/server applications'. IEEE Third Workshop on Multimedia Signal Processing, 1999, pp. 389–394
4 Hassan, S.A., Hussain, M.: 'Spatial domain lossless image data compression method'. Int. Conf. on Information and Communication Technologies (ICICT), July 2011, pp. 1–4
5 Hasan, M., Nur, K., Noor, T.B., Shakur, H.B.: 'Spatial domain lossless image compression technique by reducing overhead bits and run length coding', *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)*, 2012, **3**, (2) Q7
6 Hasan, M., Nur, K.: 'A novel spatial domain lossless image compression scheme', *Int. J. Comput. Appl.*, 2012, **39**, (15), pp. 25–28
7 Huang, S.C., Chen, L.G., Chang, H.C.: 'A novel image compression algorithm by using Log-Exp transform'. Proc. 1999 IEEE Int. Symp. on Circuits and Systems (ISCAS'99), July 1999, vol. 4, pp. 17–20
8 Rabbani, M., Jones, P.W.: 'Adaptive DPCM', in 'Digital image compression techniques' (SPIE Press, 1991) Q8
9 Weinberger, M.J., Seroussi, G., Sapiro, G.: 'The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS', *IEEE Trans. Image Process.*, 2000, **9**, (8), pp. 1309–1324
10 'Kodak lossless true color image suite', http://www.r0k.us/graphics/kodak/, accessed January 2014
11 'USC-SIPI image database', http://www.sipi.usc.edu/database/database.php?volume=misc, accessed January 2014
12 Christopoulos, C.A., Askelof, J., Larsson, M.: 'Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard', *IEEE Signal Process. Lett.*, 2000, **7**, (9), pp. 247–249
13 Moreno, J.M.: 'Perceptual criteria on image compression'. Ph.D. dissertation, DCC, UAB, Barcelona, Spain, 2011
14 Zhang, C.N., Wu, X.: 'A hybrid approach of wavelet packet and directional decomposition for imagecompression'. IEEE Canadian Conf. on Electrical and Computer Engineering, Edmonton, Alta, Canada, December 1999, vol. 2, pp. 755–760
15 Zhou, Z., Venetsanopoulos, A.N.: 'Morphological methods in image coding'. IEEE Int. Conf. on Acoustics, Speech, and Signal Process (ICASSP-92), San Francisco, CA, USA, March 1992, vol. 3, pp. 481–484
16 Popovici, I., Withers, W.D.: 'Locating edges and removing ringing artifacts in JPEG images by frequency-domain analysis', *IEEE Trans. Image Process.*, 2007, **16**, (5), pp. 1470–1474
17 Mittal, A., Moorthy, A.K., Bovik, A.C.: 'No-reference image quality assessment in the spatial domain', *IEEE Trans. Image Process.*, 2012, **21**, (12), pp. 4695–4708
18 Zhang, L., Zhang, D., Mou, X., Zhang, L.: 'FSIM: a feature similarity index for image quality assessment', *IEEE Trans. Image Process.*, 2011, **20**, (8), pp. 2378–2386
19 Kok, C.W.: 'Fast algorithm for computing discrete cosine transform', *IEEE Trans. Signal Process.*, 1997, **45**, (3), pp. 757–760

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

IPR20140421

Author Queries

Jose Javier García Aranda, Marina González Casquete, Mario Cao Cueto, Joaquín Navarro Salmerón, Francisco González Vidal

Q1 Please chek the e-mail id of the corresponding author.

Q2 We have inserted a citation for Figures 4, 5, 7 and 8. Please approve or provide an alternative.

Q3 Please check and confirm the citation of Tables 2 and 3 in the text.

Q4 We have inserted a main caption for Figure 9 as per IET style. Please approve or provide an alternative.

Q5 We have changed the algorithm to figure 9 and figures have been renumbered accordingly as computer coding wider than single column is not allowed as per style. Please check and approve.

Q6 We have changed the citation of Fig. 9 to Fig. 11. Please check and approve.

Q7 Please provide page numbers in Ref. [5].

Q8 Please provide editor names for Ref. [8].